

CLAIMS

1. A method for testing a user interface of an administration software of a cluster, said cluster comprising at least one node, said cluster administration software running on said at least one node, said cluster comprising an administrative web server running on said at least one node and providing said user interface for said cluster administration software, said method comprising:

(1) pre-recording a sequence of requests to be sent to said administrative web server of said cluster at a later time;

(2) sending said pre-recorded sequence of requests from a test node to said administrative web server of said cluster; and

(3) receiving a sequence of responses of said administrative web server of said cluster to said sent pre-recorded sequence of requests; wherein said sequence of received responses is indicative of correctness of operation of said user interface.

2. The method of claim 1, further comprising:

(4) communicating with one or more nodes of said cluster and performing direct examination of said one or more nodes; wherein results of said direct examination are indicative of correctness of operation of said user interface.

3. The method of claim 1, wherein said receiving said sequence of responses further comprises comparing said sequence of received responses with a sequence of expected responses.

4. The method of claim 3, wherein said sequence of expected responses comprises a subset of responses attributable to correct operation of said user interface.

5. The method of claim 3, wherein said sequence of expected responses comprises at least one pre-determined response.

6. The method of claim 1, further comprising:

(5) communicating with one or more applications running on said cluster to verify that said one or more applications are operating correctly.

7. The method of claim 1, wherein said pre-recording a sequence of requests comprises:

(a) requesting a test creator to input an administrative task; and
(b) storing said sequence of requests in a request file; said sequence of requests being determined by the administrative task input by said test creator.

8. The method of claim 7, further comprising:

(c) storing a sequence of expected responses in a response file, wherein each of said sequence of expected responses corresponds to at least one request from said sequence of requests stored in said request file.

9. The method of claim 8, further comprising:

(d) providing said test creator with an opportunity to examine said sequence of requests and said sequence of responses and mark said pre-recorded sequence of requests as failing.

10. The method of claim 7, further comprising:

(c) providing said test creator with an opportunity to add any additional operation code.

11. The method of claim 7, further comprising:

(c) providing said test creator with an opportunity to add additional cleanup steps, said additional cleanup steps ensuring that said at least one node of said cluster are placed in a consistent state after said testing is completed.

12. A computer-readable medium embodying a program for testing a user interface of an administration software of a cluster, said cluster comprising at least one node, said cluster administration software running on said at least one node, said cluster comprising an administrative web server running on said at least one node and providing said user interface for said cluster administration software, said program comprising:

(1) pre-recording a sequence of requests to be sent to said administrative web server of said cluster at a later time;

(2) sending said pre-recorded sequence of requests from a test node to said administrative web server of said cluster; and

(3) receiving a sequence of responses of said administrative web server of said cluster to said sent pre-recorded sequence of requests; wherein said sequence of received responses is indicative of correctness of operation of said user interface.

13. The computer-readable medium of claim 12, wherein said program further comprises:

(4) communicating with one or more nodes of said cluster and performing direct examination of said one or more nodes; wherein results of said direct examination are indicative of correctness of operation of said user interface.

14. The computer-readable medium of claim 12, wherein said receiving said sequence of responses further comprises comparing said sequence of received responses with a sequence of expected responses.

15. The computer-readable medium of claim 14, wherein said sequence of expected responses comprises a subset of responses attributable to correct operation of said user interface.

16. The computer-readable medium of claim 14, wherein said sequence of expected responses comprises at least one pre-determined response.

17. The computer-readable medium of claim 12, wherein said program further comprises:

(5) communicating with one or more applications running on said cluster to verify that said one or more applications are operating correctly.

18. The computer-readable medium of claim 12, wherein said pre-recording a sequence of requests comprises:

- (a) requesting a test creator to input an administrative task; and
- (b) storing said sequence of requests in a request file; said sequence of requests being determined by the administrative task input by said test creator.

19. The computer-readable medium of claim 18, wherein said program further comprises:

- (c) storing a sequence of expected responses in a response file, wherein each of said sequence of expected responses corresponds to at least one request from said sequence of requests stored in said request file.

20. The computer-readable medium of claim 19, wherein said program further comprises:

- (d) providing said test creator with an opportunity to examine said sequence of requests and said sequence of responses and mark said pre-recorded sequence of requests as failing.

21. The computer-readable medium of claim 18, wherein said program further comprises:

- (c) providing said test creator with an opportunity to add any additional operation code.

22. The computer-readable medium of claim 18, wherein said program further comprises:

(c) providing said test creator with an opportunity to add additional cleanup steps, said additional cleanup steps ensuring that said at least one node of said cluster are placed in a consistent state after said testing is completed.

23. A computer system comprising at least a central processing unit and a memory, said memory storing a program for testing a user interface of an administration software of a cluster, said cluster comprising at least one node, said cluster administration software running on said at least one node, said cluster comprising an administrative web server running on said at least one node and providing said user interface for said cluster administration software, said program comprising:

(1) pre-recording a sequence of requests to be sent to said administrative web server of said cluster at a later time;

(2) sending said pre-recorded sequence of requests from a test node to said administrative web server of said cluster; and

(3) receiving a sequence of responses of said administrative web server of said cluster to said sent pre-recorded sequence of requests; wherein said sequence of received responses is indicative of correctness of operation of said user interface.

24. The computer system of claim 23, wherein said program further comprises:

(4) communicating with one or more nodes of said cluster and performing direct examination of said one or more nodes; wherein results of said direct examination are indicative of correctness of operation of said user interface.

25. The computer system of claim 23, wherein said receiving said sequence of responses further comprises comparing said sequence of received responses with a sequence of expected responses.

26. The computer system of claim 25, wherein said sequence of expected responses comprises a subset of responses attributable to correct operation of said user interface.

27. The computer system of claim 25, wherein said sequence of expected responses comprises at least one pre-determined response.

28. The computer system of claim 23, wherein said program further comprises:

(5) communicating with one or more applications running on said cluster to verify that said one or more applications are operating correctly.

29. The computer system of claim 23, wherein said pre-recording a sequence of requests comprises:

(a) requesting a test creator to input an administrative task; and
(b) storing said sequence of requests in a request file; said sequence of requests being determined by the administrative task input by said test creator.

30. The computer system of claim 29, wherein said program further comprises:

(c) storing a sequence of expected responses in a response file, wherein each of said sequence of expected responses corresponds to at least one request from said sequence of requests stored in said request file.

31. The computer system of claim 30, wherein said program further comprises:

(d) providing said test creator with an opportunity to examine said sequence of requests and said sequence of responses and mark said pre-recorded sequence of requests as failing.

32. The computer system of claim 29, wherein said program further comprises:

(c) providing said test creator with an opportunity to add any additional operation code.

33. The computer system of claim 29, wherein said program further comprises:

(c) providing said test creator with an opportunity to add additional cleanup steps, said additional cleanup steps ensuring that said at least one node of said cluster are placed in a consistent state after said testing is completed.